BUILD A WIDE-BUS TEST AND DEVELOPMENT SYSTEM

DESIGNERS FACE UNIQUE CONSTRAINTS WHEN USING OFF-THE-SHELF SCSI PROTOCOL CHIPS IN SCSI-2 SYSTEMS.

MICHAEL NGUYEN
Adaptec Inc., 691 S. Milpitas Blvd., Milpitas, CA 95035; (408) 945-8600.
MARK S. GORDON
Digital Finesse Inc., 355 W. Olive Ave., Suite 105, Sunnyvale, CA 94086; (408) 737-9156.

One of the key improvements in the proposed SCSI-2 standard is a wider data bus—16- and 32-bit capability in addition to the current SCSI standard's 8-bit data width. But the way that data is sent—over two cables that don't need synchronization—leads to potential problems in debugging new designs. One solution to these problems is a test-and-development system that displays bus phases, data-bus and control-line content, and timing information. With such a system, designers wouldn't need to decode ones and zeros to understand what's happening on the bus. Instead, they could concentrate on *why* a condition occurs rather than how to test for it.

Such SCSI test-and-development systems are usually based on protocol chips specifically designed to interface with the SCSI bus. These devices contain state machines that automatically handle all bus phases and the accompanying data transfers. The protocol chips currently available, however, were designed to conform with the SCSI-1 standard. To build a SCSI-2 test system with these devices, designers must add hardware and software that support the features embodied in SCSI 2.

A board that adds wide-bus capability to the Adaptec SDS-3 SCSI test and development system illustrates some of the problems SCSI-2 system designers must face. The wide-bus option board, as it's called, employs the Adaptec AIC-6250 SCSI protocol chip. The system consists of cards that plug into an IBM PC/XT, AT, or compatible, and the software to control the cards.

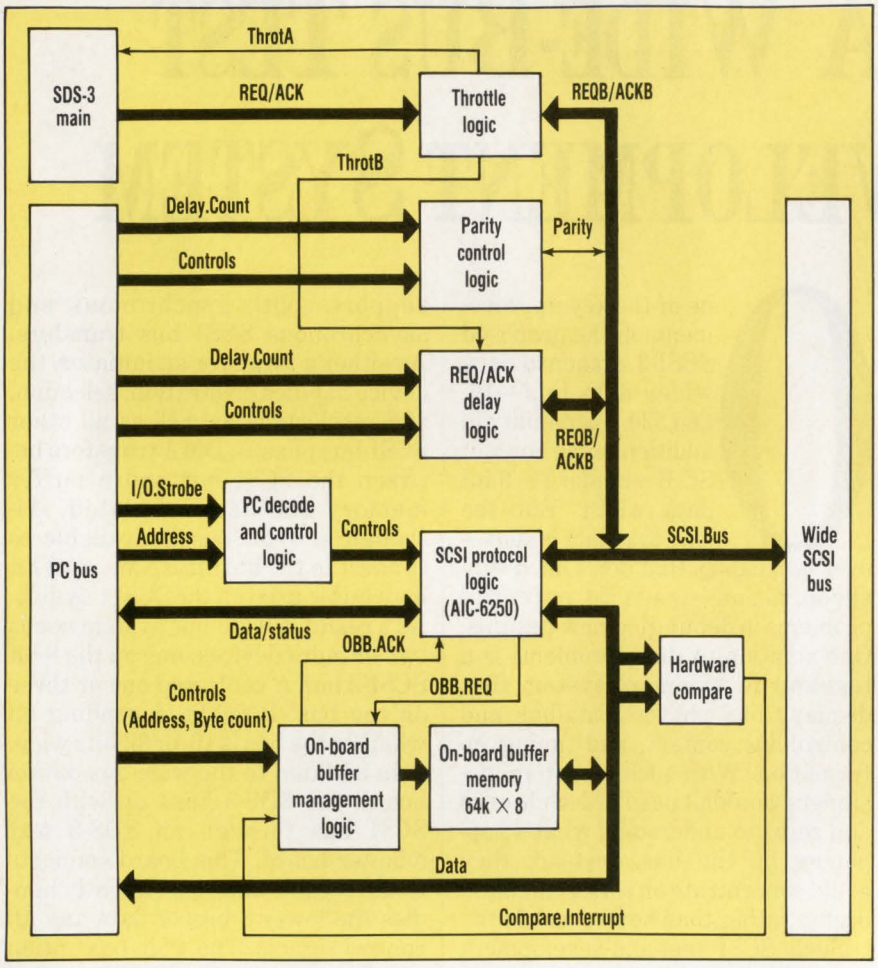
The 6250, a high-performance SCSI interface protocol controller, supports both synchronous and asynchronous SCSI bus transfers. As either a target or an initiator, the device handles arbitration, selection, and reselection, as well as all other SCSI-bus phases. DMA transfers between the SCSI bus and a buffer memory are accommodated. Although a 16-bit bus is available to connect to the buffer RAMs, an 8-bit data bus is used on the SCSI bus side. As a result, a wide-bus system needs two or more devices, one on the 8-bit SCSI-2 bus A cable and one or three on the bus' B cable, depending on whether the bus is 16- or 32-bits wide.

In addition to the wide-bus option board, the SDS-3 links up with the SCSI bus through an SDS-3 test adapter board. This board connects to the A cable and consequently handles the lower 8 bits of data and all control signals. The wide-bus option board connects to the B cable and to the test adapter board. The design example that's discussed has a 16-bit bus, so only one 6250 is needed on each board. The example also exclusively covers the wide-bus option board design.

ON-BOARD MEMORY

Besides the AIC-6250, the wide-bus option board holds an on-board data buffer (OBB memory), which stores up to 64 kbytes of data coming from or going to the SCSI bus *(Fig. 1)*. Fast DMA logic (OBB management logic) automatically performs the handshake on the OBB memory bus. This circuitry also ensures that the OBB on the wide-bus option board and the OBB on the test adapter board are synchronized so that they can transfer the correct number of bytes and locate these bytes in the buffer properly. The address and control signals from the PC bus transmit to the OBB through the

WIDE-BUS TEST SYSTEM



1. THOUGH THE AIC-6250 SCSI protocol chip was designed to conform to the original SCSI specification, designers can use the device in systems for the proposed SCSI-2 standard, such as this SDS-3 wide-bus option board.

OBB management logic.

The hardware compare block contains circuitry that compares data at high speeds. This capability makes possible on-the-fly comparisons of data that's written to a device on the SCSI bus. It works by comparing what's stored in the OBB with data that's read back from the bus (which is only compared, not stored). This feature is particularly useful when large volumes of data are transferred, making visual comparison inefficient.

The parity control logic makes controlled parity generation possible. If enabled, it generates parity for writes to the SCSI bus and checks parity on reads from the bus. With this parity logic, users can also force a parity error after from 1 to 256 byte transfers.

Because it resides on the PC bus, the board also has PC decode and control logic, which is contained in a PAL16R4, a register-type programmable logic array *(Fig. 2)*. Proper synchronization is ensured by clocking the array with the same 20-MHz on-board oscillator that the 6250 uses. The PAL translates PC bus address and control lines $\overline{IOW}$, $\overline{IOR}$, and AD_4, and address decoder output $\overline{BDSEL}$ to the $\overline{WR}$, $\overline{RD}$, ALE, and CS control lines required by the 6250.

The board has 32 (decimal) addresses: 16 for the 6250's internal registers and 16 for other logic. AD_4 chooses between the two sets, and the $\overline{BDSEL}$ line supplies access to any of the 32 locations.

Two three-state buffers, $TSBUF_1$ and $TSBUF_2$, isolate the address bus lines from the multiplexed address/data bus lines, PD_{7-0}, of the 6250. Three more three-state buffers ($TSBUF_3$, $TSINV_1$, and $TSINV_2$) connect the $\overline{SCSI.I/O}$, $\overline{Pseudo.C/D}$, and $\overline{Pseudo.BSY}$ lines to the device.

ADAPTING TO SCSI-2

The $\overline{Pseudo.C/D}$ and $\overline{Pseudo.BSY}$ signals are needed because the 6250 is a SCSI-1 type device working in a SCSI-2 wide-bus system. This is because the 6250 is designed to handle all of the SCSI bus phases, expecting to see them occur in a particular order. Because all control signals are transmitted on the A cable, the 6250 on the adapter board monitors and handles the phases that these signals specify.

On the other hand, the 6250 on the wide-bus option board handles only B cable data transactions, so it doesn't see BSY, SEL, or any of the phase lines carried on the A cable. The $\overline{Pseudo.BSY}$ and $\overline{Pseudo.C/D}$ signals make the 6250 on the wide-bus option board believe that those SCSI-bus phases preceding the data phase are complete.

As a result, the 6250 on the B cable can arbitrate, select (or reselect), watch for a response to its selection (or reselection), and detect phases other than the data phase (a command, status, or message phase) so it can properly proceed to and from the SCSI data phase. Control software that's part of the SDS-3 manages the Pseudo.BSY and Pseudo.C/D signals.

It must be remembered that the BSY, SEL, and phase control lines (C/D, I/O, and MSG) of the 6250 on the wide-bus option board aren't actually connected and this device doesn't control or respond to bus phases. The 6250 on the SDS-3 test adapter board handles bus phases.

A look at the arbitration and selection/reselection phases shows how the process works. Acting as an initiator, the 6250 on the wide-bus option board believes it's commencing arbitration when bit 6 in its interrupt mask register 0 is set to 1. The initiator raises SEL (if arbitration is won), drops BSY, and waits for the target

WIDE-BUS TEST SYSTEM

to respond to being selected by raising BSY.

The SDS-3 control software responds when the SCSI SEL signal appears by polling SCSI signal register 09 on the 6250—this determines when BSY is dropped. It then raises the Pseudo.BSY signal at the appropriate time. This particular action fools the 6250 into believing that a SCSI target responded.

No Handshake Needed

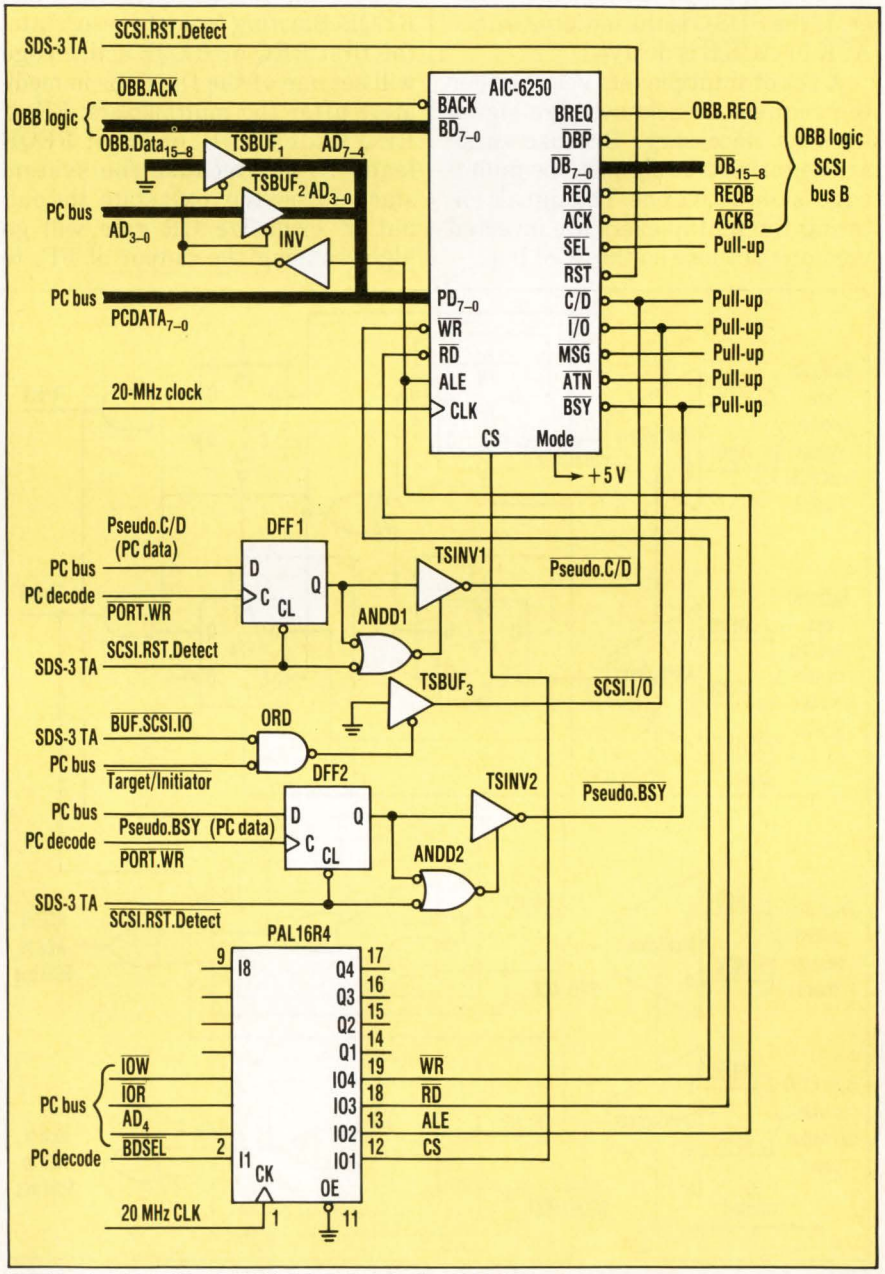
The control software then asserts the Pseudo.C/D signal to simulate a change to command phase by a target. The 6250 matches this new phase, which is reflected in the SCSI C/D bit (bit 7 of the device's SCSI signal register 09). Next, the control software drops Pseudo.C/D , forcing a change in value of the SCSI C/D bit. Note that a handshake isn't needed for the 6250 to proceed to the SCSI data phase.

Similarly, when the board acts as a target, the control software watches for SEL being raised and BSY being dropped, then asserts and deasserts the Pseudo.BSY signal. This action makes the 6250 believe that it successfully completed the arbitration and reselection phases and gained control of the SCSI bus by asserting BSY. As a target, the 6250 can be switched to the command phase by writing SCSI signal register 09, bit 7. Data phase is entered when this bit is deasserted. This series of actions places the device in the data phase so that it can transfer data properly. The SCSI control signals are different for a target and an initiator, so the Target/Initiator signal is required to differentiate between the two roles.

The SCSI.I/O signal (a buffered version of the SCSI bus I/O signal) forces the 6250 to see the data phase direction (in or out) on the SCSI bus. The SCSI.RST.Detect signal resets the 6250 and then deasserts the Pseudo.BSY and Pseudo.C/D signals when a reset occurs on the SCSI bus.

As noted, the proposed SCSI-2 standard doesn't supply synchronized data transfers on the A and B cables. In the case of a 16-bit data bus, even bytes (0, 2, 4, and so on) pass along the A cable, and odd bytes (1, 3, 5, and so on) along the B cable. If the cables are of different lengths, which is permitted by the proposed standard, skewed transfers can occur. For instance, bytes 0, 2, and 4 may pass along the A cable before the transfer of byte 1 is completed on the B cable.

A buffer, however, must hold the bytes in their proper order (0, 1, 2, and so forth), even though that may differ from the order of receipt if transfers on one cable operate faster than those on the other. Consequently, unsynchronized cables can make it difficult to debug firmware that, for example, moves data into a disk buffer on reads. If the system enforces the synchronization of the two cables—a process called throttling—designers can turn the synchronization on and off to determine if skewed transfers cause improperly ordered buffers.



2. A PROGRAMMABLE LOGIC ARRAY, the PAL16R4, translates PC bus address and control lines into the control signals required by the SCSI protocol chip.

WIDE-BUS TEST SYSTEM

Fortunately, SCSI 2 includes separate REQ and ACK lines on each cable to accommodate possible different cable lengths. Therefore, it's conceptually easy to throttle transfers on either the A or B cable so that they're within one byte of each other.

To enforce synchronization, a throttle logic circuit generates two signals, ThrotA and ThrotB, that delay the appropriate $\overline{\text{REQ}}$ or $\overline{\text{ACK}}$ signal *(Fig. 3)*. If the SDS-3 is the target, either $\overline{\text{REQ}}$ or $\overline{\text{REQB}}$ is delayed, depending on which cable leads the other. If the SDS-3 is the initiator, either $\overline{\text{ACK}}$ or $\overline{\text{ACKB}}$ is delayed.

A set of multiplexers select either the request or acknowledge signal lines as necessary. Because these are negative true signals, the falling edge is the active one. The signals entering the multiplexers are inverted versions of those on the SCSI bus.

Four D flip-flops then determine which line (A or B) leads the other by more than one active edge. The Throttle signals pass through AND-OR logic that delays assertion on the SCSI bus of the appropriate handshake signal from the 6250 until it's less than one active edge ahead of its counterpart.

THROTTLING ACTION

For example, assume that the SDS-3 is the target and is therefore responsible for asserting REQ and REQB. Starting from the reset state, the first REQ or REQB active edge will set one of the flip-flops immediately after the multiplexers: FF_1 if REQ leads REQB, or FF_2 if REQB leads REQ. Because the system started from the reset state, the output of exclusive OR gate will go high, causing the output of FF_3 to also go high. This output, combined with the enclusive OR gate's output, activates the ThrotA signal.

ThrotA, in turn, is an input to the AND-OR circuit that throttles the A cable. Working as a latch, this circuit waits until the current handshake transaction is completed, which occurs when $\overline{\text{REQ}}$ in the example goes high or inactive. This high signal combines with the high ThrotA to drive DelayA high, preventing $\overline{\text{REQ}}$ from being asserted at the OR gate's output.

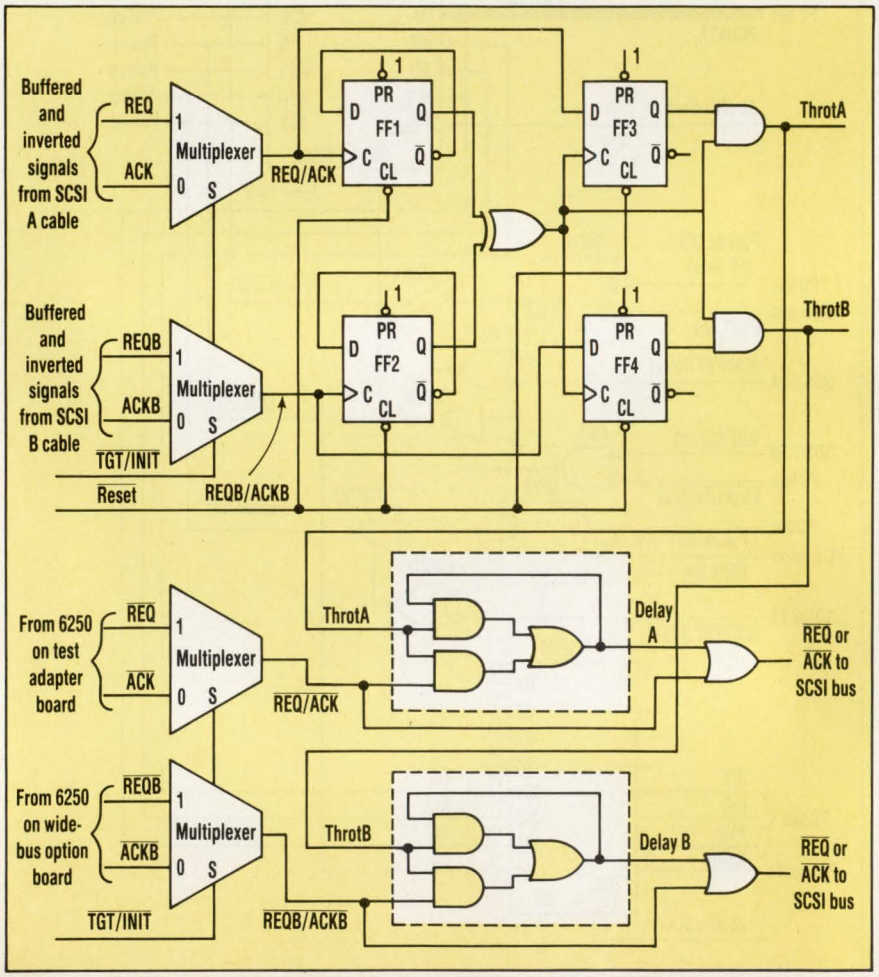
Because DelayA is fed back into the AND-OR network, the circuit is latched until ThrotA goes low, regardless of the state of $\overline{\text{REQ}}$. And ThrotA can't go low or inactive until an active edge occurs on the REQB line. The result is the desired throttling: $\overline{\text{REQ}}$ can't have two active edges without an intervening active $\overline{\text{REQB}}$ edge.

When the next $\overline{\text{REQB}}$ edge does occur, it triggers flip-flop FF_2. The exclusive OR gate's output then goes low and negates ThrotA. With ThrotA negated, another $\overline{\text{REQ}}$ edge can be gated onto the SCSI bus, making it possible for another data transfer on the A cable.

A similar series of events occurs if the $\overline{\text{REQB}}$ signal attempts to precede the $\overline{\text{REQ}}$ signal by more than one active edge. In that case, the ThrotB signal is activated and the B cable transfer throttled. The same situation applies if the SDS-3 is the initiator. The only difference between these situations is that the $\overline{\text{ACK}}$ or $\overline{\text{ACKB}}$ signal is throttled.

By blocking another REQ active edge generation, this technique prevents more data transfers from occurring. As a result, the REQ/ACK handshaking ensures that transfers occur in lock step. This method is used only with asynchronous SCSI transfers.

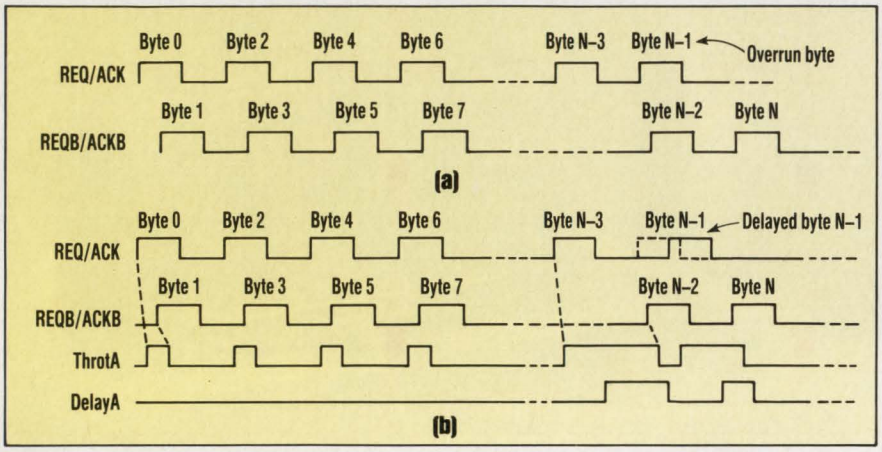
The affect of throttling is easily seen on a set of typical REQ/ACK signal timing diagrams. With the throttling circuit disabled, byte N–1 is an "overrun byte," which means that its transfer begins before the transfer of byte N–2 begins *(Fig. 4a)*. Debugging this situation can be difficult.



3. BY ADDING THROTTLING LOGIC on the wide-bus board, designers can synchronize A and B cable operation for easier debugging of SCSI 2 designs.

WIDE-BUS TEST SYSTEM

With the throttling circuit working, the active edges of the REQ/ACK and REQB/ACKB signals control the state of ThrotA. The ThrotA and DelayA signals delay the byte N–1 transfer *(Fig. 4b)*. ThrotA goes high when the REQ/ACK signal associated with byte N–3 goes high. DelayA goes high when REQ/ACK goes in active. The DelayA signal prevents the REQ/ACK signal, gated onto the SCSI bus by the OR gate in the throttling circuit, from going active until the next active edge of REQB/ACKB arrives. That next active edge of REQB/ACKB causes ThrotA to go low, which forces DelayA low.□



4. WITH THE THROTTLING FEATURE OFF, the transfer of byte N-1 may begin before the transfer of N-2 (a). Enabling the throttling circuit makes it possible for the REQ/ACK lines to control the Throttle and Delay signals in a way that delays the transmission of N-1 until N-2 begins transmitting (b).

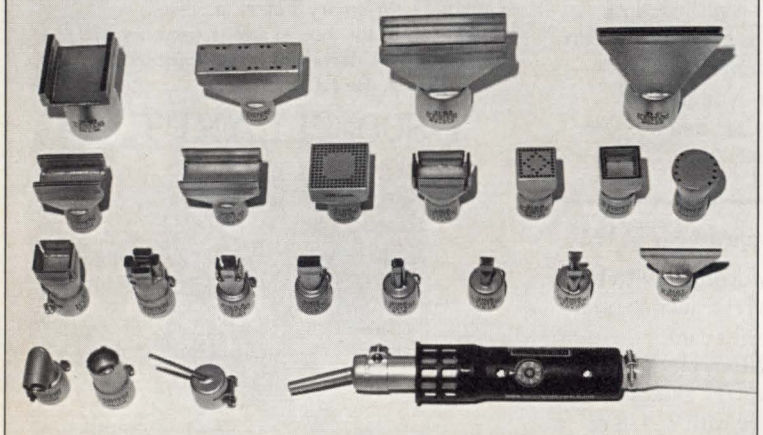
Michael Nguyen, an engineer in the Development System Operations group at Adaptec, previously developed hardware at Link Flight Simulation.

Mark S. Gordon, founder and president of Digital Finesse, a consulting-management company, designed ICs at Zilog and helped found Verticom Inc.

HOW VALUABLE?	CIRCLE
HIGHLY	544
MODERATELY	545
SLIGHTLY	546

Contact-free soldering and de-soldering

of SMD, DIP and pin grid components and connectors in a matter of seconds with the Leister-Labor "S" hot-air tool. Electronic control of temperature and air supply. Over 400 special nozzles available.



Ask for free brochure UW 169

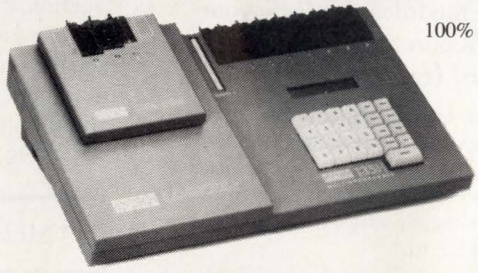
Brian R. White Co. Inc., 313 Henry Station Road
Ukiah, CA 95482 phone: (707) 462-9795
Farmingdale, NJ 07727 phone: (201) 938-2700

CIRCLE 105

BYTEK 135H

Universal MultiProgrammer™

DATA I/O 288* Performance
PLUS PLD Support !!



100% USA Made.

Model Shown: 135H-U with optional LOGICel.

★ Model 135H is a *GANG & SET (E)EPROM* Programmer.
★ Programs virtually all EPROMs & EEPROMs up to 2048K.
★ Universal support includes Bipolar PROMs & EPROM Emulation.
★ ***GANG Options for 40-pin MICROs & 40-pin EPROMs !!***
★ Stand-alone or Computer Remote Control operation.
★ 12 Month WARRANTY & 12 Month **FREE** Field Device Updates.
★ Supports OrCAD™'s new PLD Tools Software.

BYTEK **1-800-523-1565**
In Florida: (407) 994-3520

BYTEK Corporation • 508 N.W. 77th St, Boca Raton, FL 33487
FAX: (407) 994-3615 • Telex: 4998369 BYTEK

* DATA I/O is a registered trademark of DATA I/O Corporation.

CIRCLE 106